Categorical Semantics for a Quantum Language*

Martin Churchill
Joint Work with Samson Abramsky

July 21, 2010

Abstract

We prove a correspondence theorem for a quantum programming language in an axiomatic (categorical) setting. We present a simple while-based programming language for a machine that has access to Quantum Systems (in particular a system of *qubits*) and the relevant operations on them. We give (coinciding) operational and denotational semantics for this language at a concrete level (Hilbert spaces and linear maps) and then proceed to do so at a more abstract categorical level, using ideas from [AC04] and [Sel07].

1 Introduction

Quantum Mechanics has traditionally been axiomatised in terms of Hilbert Spaces (a *qubit* can be represented by a pair of complex numbers up to non-zero multiple — a ray in the space $\mathbb{C}^2$). Combination of systems results in the possibility of quantum entanglement, modelled by the tensor product. In recent years, it has been noticed that the structure required for reasoning about quantum systems can be reduced to a small number of categorical axioms. Thus in [AC04] Abramsky and Coecke recast Quantum Mechanics into a new light — we no longer need a Hilbert Space, but something weaker: a certain type of category that happens to support the quantum features we require (and in particular, the category of sets and relations also admit these features).

Within this new framework, many derived concepts from Quantum Mechanics (and general Linear Algebra) are definable. For example, it is possible to define and prove the teleportation protocol at this level of abstraction [AC04]. Further, this new formulation directly gives rise to a graphical calculus, i.e. a typed high-level way of reasoning about quantum computation; something that was very much missing (and missed) in Quantum Mechanics of the 20th century. These "grapical" proofs have a formal foundation [JS91, Sel04a]. The fundamental two-dimensionality comes from the orthogonal notions of composition (time) and tensor product (space).

The graphical calculus attempts to address the problem of a lack of high-level quantum formulation. This issue certainly needs addressing — development of quantum algorithms tends to depend on mathematical tricks and hacking

*This is a short writeup of the main result from my masters thesis, supervised by Samson Abramsky. Perhaps this will eventually be submitted somewhere.

matrices. As such, development of quantum algorithms has been *ad hoc* and rare. Another way of dealing with this is by considering programming languages for quantum systems. One successful endeavour in this area has been the work of Peter Selinger — in [Sel04b] a quantum language is presented, along with some (denotational) semantics for that language in terms of Hilbert spaces. For a review of the status of quantum programming languages, see [Sel04a]. Here we consider a simple while-based language presented in [Abr04]. We shall unify the concrete exposition of this language with the categorical axiomatics of [AC04]. We firstly present the concrete version of the language and its semantics; and then generalise to the categorical setting. We will do this both operationally and denotationally, and give a correspondence result between the two. As far as we know, this is the first result of this kind using the axiomatic, categorical approach to quantum mechanics.

The language in question is a simple `while` language for a classical computer with access to quantum systems and quantum operations. This language consists of imperative constructs — sequencing, iteration and conditioning — together with commands for updating the classical state space, and applying quantum operations to the qubits (including measurement, leaving the result in a classical register).

2 A Quantum Language

2.1 QRAM Machines

We briefly recall some quantum formalisms. The quantum state of a particle can be represented as a point on a sphere. This corresponds to a pair of complex numbers up to non-zero complex multiple, i.e. a ray in a two-dimensional Hilbert space. This can be generalised to multiple qubits and so we define the state space of a quantum system to be a finite-dimensional Hilbert space H. A state within this state space corresponds to a ray (i.e. a one dimensional subspace) of H, typically represented by a vector of unit norm.

So the state of a qubit (as a base case for quantum system) can be represented by a ray in $\mathbb{C}^2$. To combine to quantum systems we use the tensor product of the two Hilbert spaces (rather than the direct sum/product of the space sets). This is a key component of the quantum exponential speed-up — since the dimensions grow exponentially rather than linearly as further systems are added.

We cannot read and write to quantum bits arbitrarily. The only operations we can perform on quantum systems are *unitary* ones (adjoint = inverse). In the qubit case, this corresponds precisely to rotations. In particular all of these operations are invertible.

For reading qubits, we must perform *quantum measurements*. Given some qubit q in state $\alpha_0|0\rangle + \alpha_1|1\rangle$ (for $\alpha_0, \alpha_1 \in \mathbb{C}$) we can choose to ask the question of whether the state of q is $|0\rangle$ or $|1\rangle$ (of course, it may be in neither). We will get answer to this (by measuring the qubit) that will be either a $|0\rangle$ or $|1\rangle$, and which it is will depend *probabilistically* on whether the state of q is "closer" to being in state $|0\rangle$ or $|1\rangle$. This closeness measure is performed using the inner product, which in the case of qubits on the surface of a sphere gives the intuitive result. Furthermore, if the result of the measurement was that the state is closer to $|0\rangle$ than $|1\rangle$, the state of q *will become* $|0\rangle$, and vice versa. Thus the act of

measuring the qubit destroys the state of the qubit. We can generalise this, measuring the state of the system with respect to any basis. Given each vector in the basis b_i the state of the system becomes b_i with probability $\langle b_i|q\rangle$ (making use here of the inner product,) we are informed of the resultant state.

So quantum systems are inherently different to classical systems — they have a continuous state space, that exponentially grows, but we can't read them as we'd ideally like to and without strange consequence. This is what makes writing quantum algorithms hard as mentioned above, and why developing quantum algorithms is akin to 'hacking around with complex numbers'. We define a *QRAM machine* to be a classical machine together with access to a quantum system — that is, some number of qubits and the ability to perform the operations described on them, as in [Sel04a]. We next briefly describe a basic procedural scripting-language around these quantum operations.

2.2 An Imperative Language

We define a programming language for a computer that has access to some number of quantum bits, and that can perform the above operations on those qubits. We assume a syntax of arithmetic expressions aexp, Boolean expressions bexp, unitary quantum primitives uni; as well as quantum and classical program variables qvar and var. It can be shown that from some small basis of binary unitary operators we can define all unitary operators, and that we only need to be able to measure with respect to a standard basis to be able to measure with respect to any [NC00]. Hence, restricting unitaries to those available in uni and measurements to the computational basis does not restrict expressivity.

We then define the syntax of commands as follows:

C ::= var := aexp | skip | C_1;C_2 | if bexp then C_1 else C_2 | while bexp do C | apply uni to qvar, qvar | measure qvar in var

Before giving categorical semantics for this language, we first briefly consider concrete operational and denotational semantics. Let L be the set of values that a variable can store, and S be the set of mappings from var into L — our classical state space. We assume a QRAM machine of k qubits, and our quantum state space H will be given by the Hilbert space $Q^{\otimes k} = Q \otimes \ldots \otimes Q$ where Q is our qubit space $\mathbb{C}^2$. We can consider a basis for H given by the set of binary strings of length k, and if b is such a string we will write $|b\rangle$ for the corresponding basis vector. Configurations are triples (C, s, ϕ) where C is a command, $s \in S$ and ϕ is a ray in H.

We define a reduction relation on configurations, but since this reduction relation is probabilistic (due to quantum measurements) each reduction must be assigned a probability p — we write $(C, s, \phi) \rightarrow^p (C', s', \phi')$. We will have a further coherence condition stating that if (C, s, ϕ) reduces to one of many possibilities, the probabilities add up to (at most) one. We also require that the nondeterminism is countable, i.e. each configuration reduces to only countably many configurations (this is clearly true of the reduction relation below). We assume "primitive" semantics for aexps $[\![e]\!] : S \to L$, bexps $[\![b]\!] : S \to \{\mathsf{tt}, \mathsf{ff}\}$ and unis $[\![U]\!] : Q \otimes Q \to Q \otimes Q$ where S is our set of classical configurations. We give our reduction relation as follows:

$$(v := e,\, s,\, \phi) \to^1 (\mathtt{skip},\, s[v \mapsto [\![e]\!](s)],\, \phi)$$

$$(\mathtt{skip};C,\, s,\, \phi) \to^1 (C,\, s,\, \phi)$$

$$\frac{(C_1,\, s,\, \phi) \to^p (C_1',\, s',\, \phi')}{(C_1;\, C_2,\, s,\, \phi) \to^p (C_1';\, C_2,\, s',\, \phi')}$$

$$\frac{[\![b]\!](s) = \mathsf{tt}}{(\mathtt{if}\ b\ \mathtt{then}\ C_1\ \mathtt{else}\ C_2,\, s,\, \phi) \to (C_1,\, s,\, \phi)}$$

$$\frac{[\![b]\!](s) = \mathsf{ff}}{(\mathtt{if}\ b\ \mathtt{then}\ C_1\ \mathtt{else}\ C_2,\, s,\, \phi) \to (C_2,\, s,\, \phi)}$$

$$\frac{[\![b]\!](s) = \mathsf{tt}}{(\mathtt{while}\ b\ \mathtt{do}\ C,\, s,\, \phi) \to (C;\, \mathtt{while}\ b\ \mathtt{do}\ C,\, s,\, \phi)}$$

$$\frac{[\![b]\!](s) = \mathsf{ff}}{(\mathtt{while}\ b\ \mathtt{do}\ C,\, s,\, \phi) \to (\mathtt{skip},\, s,\, \phi)}$$

$$(\mathtt{apply}\ U\ \text{to}\ q_1, q_2,\, s,\, \phi) \to (\mathtt{skip}, s, [\![U]\!]_{i,j}(\phi))$$

$$(\mathtt{measure}\ q\ \mathtt{in}\ v,\, s,\, \phi) \to^{p_0^q(\phi)} (\mathtt{skip},\, s[v \mapsto 0],\, P_0^q(\phi))$$

$$(\mathtt{measure}\ q\ \mathtt{in}\ v,\, s,\, \phi) \to^{p_1^q(\phi)} (\mathtt{skip},\, s[v \mapsto 1],\, P_1^q(\phi))$$

Here for each quantum state $\phi \in Q^k$ with $\phi = \Sigma \alpha_b |b\rangle$ we define $p_j^i(\phi) = \Sigma \left\{ |\alpha_b|^2 : b_i = j \right\}$ and $P_j^i(\phi) = \frac{1}{\sqrt{p_j^i}}.\Sigma \left\{ \alpha_b |b\rangle : b_i = j \right\}$. Thus $p_j^i(\phi)$ is the probability that we will get the result j on measuring the ith qubit in state ϕ and $P_j^i(\phi)$ is the collapsed state that will result from such a measurement. If U is a unitary, we define $[\![U]\!]_{i,j}$ to be the operation $H \to H$ applying $[\![U]\!]$ to qubits i and j (so this is of the form $\sigma^{-1} \circ \mathsf{id} \otimes U \circ \sigma$ for an appropriate symmetry isomorphism σ).

Define $\mathsf{DProb}(A)$ to be the set of discrete probability sub-distributions on A, i.e. mappings $p : A \to [0, 1]$ such that $d_p = \{b \in A : p(b) \neq 0\}$ is countable with $\Sigma_{a \in d_p} p(a) \leq 1$. Given any command C we define the operational meaning of that command $\mathcal{O}(C)$ as a function $S \times H \to \mathsf{DProb}(S \times H)$. Given a command C together with $(s, \phi) \in S \times H$ we can construct a reduction tree starting from (C, s, ϕ) in the obvious manner. To find the resulting distribution on $S \times H$ we set all probabilities to zero except those that exist as a leaf at the base of the reduction tree. These configuration will be precisely those of the form $(\mathtt{skip}, s', \phi')$. For such leaves, with $(C, s, \phi) \to^{p_1} \ldots \to^{p_n} (\mathtt{skip}, s', \phi')$ their probability in the distribution will be Πp_i. If the configuration $(\mathtt{skip}, s', \phi')$ exists as multiple leaves in the reduction tree, the sum of the associated probabilities is taken.

As well as defining the operational semantics we can directly define a compositional denotational semantics by giving meanings of commands C directly as $\mathcal{D}(C) : S \times H \to \mathsf{DProb}(S \times H)$. Given $f : A \to B$ we define $\widehat{f} : A \to \mathsf{DProb}(B)$

Table 1: Concrete Semantics

$$
\begin{array}{lcl}
\mathcal{D}[\![\texttt{skip}]\!] & = & \widehat{\mathsf{id}} \\
\mathcal{D}[\![v := e]\!] & = & \widehat{f} \text{ where } f = \lambda(s,\phi).(s[v \mapsto [\![e]\!](s)],\phi) \\
\mathcal{D}[\![C_1; C_2]\!](s,s'') & = & \Sigma_{s'}(\mathcal{D}[\![C_1]\!](s,s').\mathcal{D}[\![C_2]\!](s',s'')) \\
\mathcal{D}[\![\texttt{if } b \texttt{ then } C_1 \texttt{ else } C_2]\!](s,\phi) & = & \mathsf{if}\ [\![b]\!](s) = \mathsf{tt}\ \mathsf{then}\ \mathcal{D}[\![C_1]\!](s,\phi)\ \mathsf{else}\ \mathcal{D}[\![C_2]\!](s,\phi) \\
\mathcal{D}[\![\texttt{while } b \texttt{ do } C]\!] & = & \mathsf{lfp}[\lambda f : S \times H \to \mathsf{DProb}(S \times H).\lambda(s,\phi) : S. \\
 & & \mathsf{if}[\![b]\!](s) = \mathsf{tt}\ \mathsf{then}\ (\mathcal{D}[\![C]\!]; f)(s,\phi)\ \mathsf{else}\ \widehat{id}(s,\phi)] \\
\mathcal{D}[\![\texttt{apply } U \texttt{ to } q_1, q_2]\!](s,\phi)(s',\phi') & = & \begin{cases} 1 & \text{if } s = s' \wedge \phi' = U_{i,j}(\phi) \\ 0 & \text{otherwise} \end{cases} \\
\mathcal{D}[\![\texttt{measure } q \texttt{ in } v]\!](s,\phi)(s',\phi') & = & \begin{cases} p_0^q(\phi) & \text{if } s' = s\,[v \mapsto 0] \wedge \phi' = P_0^q(\phi) \\ p_1^q(\phi) & \text{if } s' = s\,[v \mapsto 1] \wedge \phi' = P_1^q(\phi) \\ 0 & \text{otherwise} \end{cases}
\end{array}
$$

in the obvious manner with $\widehat{f}(a)$ having precisely one nonzero probability at $f(a)$ whose probability is 1. We define $\mathcal{D}(C)$ in table 1.

In the `while` case, we note that $\mathsf{DProb}(A)$ can be made into a domain with functions ordered pointwise in the obvious manner (thus a probability of 0 is the undefined element). This justifies our fixpoint construction. Note that there are some (potentially) infinite sums in the above descriptions, but we can use countability to define them and boundedness to justify their convergence. We prove the following in [Chu07]:

Proposition 2.1 *For any command C, $\mathcal{O}(C) = \mathcal{D}(C)$.*

2.3 Example: Quantum Teleportation

We can give an example of a program expressed in this language. We can express the protocol for *quantum teleportation* locally, which can be seen as moving the value (status) of one quantum variable a (qubit) to the status of another b. To do this, one needs an extra qubit c. The algorithm is as follows [BBC$^+$93]:

- Set b and c to be jointly in the state $|00\rangle + |11\rangle$
- Measure a and b with respect to the *Bell basis* (this destroys the state of a).
- Depending on one of the four results, apply a unitary to c.

It is well known that this sets the state of c to the original state of a. This algorithm can be represented as a program in this quantum language. We describe how each of the above three steps can be implemented.

For the first, we assume that the matrix U_B given by

$$
\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 \\ 1 & -1 & 0 & 0 \end{pmatrix}
$$

is one of our unitaries. We then note that $|00\rangle + |11\rangle = U_B(|00\rangle)$ and so reduce the problem of setting b and c to $|00\rangle$, i.e. each of b and c to $|0\rangle$. This is possible by performing a (computational base) measurement, and in the case of resulting in $|1\rangle$ applying a rotation matrix R sending $|1\rangle$ to $|0\rangle$.

We then measure a and b with respect to the Bell basis. It is sufficient to use U_B to perform a change of basis and measure in the computational base. So we can apply $U_B^\dagger$ to a and b, then measure (leaving the results in v_1 and v_2, then apply U_B to a and b. This is actually two measurements (of a and b respectively) leaving us in one of four states depending on the values of v_1 and v_2. We then apply one of four different operators to b depending on the values of v_1 and v_2 using the conditional. We assume each of these rotations are available in our set of unitaries; if not it will be possible to construct them.

So the complete program is

```
measure b in v1; if v1 = 1 then apply rot180 to b;
measure c in v1; if v1 = 1 then apply rot180 to c;
apply U_B† to a, b;
measure a in v1; measure b in v2;
apply U_B to a, b;
if v1 = 0 and v2 = 0 then apply U00 to c else skip;
if v1 = 0 and v2 = 1 then apply U01 to c else skip;
if v1 = 1 and v2 = 0 then apply U10 to c else skip;
if v1 = 1 and v2 = 1 then apply U11 to c else skip
```

The semantics of this program gives the correct "quantum teleportation" behaviour, by expanding its semantics and standard reasoning. In [Chu07] we perform a more detailed analysis for the Deutch-Josza algorithm (and with respect to the categorical categorical semantics presented below).

3 Categorical Semantics

3.1 Semantic Preliminaries

We now define the semantics of the language with respect to the categorical formulation of quantum mechanics developed in [AC04]. In particular quantum mechanics can be formulated in any *strongly compact closed category with biproducts*, of which the usual **FdHilb** is an example (another is the category **Rel**). The following definition is given in [AC05]:

Definition A *strongly compact closed category with biproducts* (SCCCB) is a symmetric monoidal category (writing $\sigma_{A,B}$ for the symmetry isomorphism $A \otimes B \cong B \otimes A$) equipped with: A monoidal involutive assignment $A \mapsto A^*$ on objects; an identity-on-objects, contravariant, monoidal, involutive functor $f \mapsto f^\dagger$; and a unit $\epsilon_A : I \to A^* \otimes A$ with $\epsilon_{A^*} = \sigma_{A^*,A} \circ \epsilon_A$; and a biproduct structure; such that $(\epsilon_A^\dagger \circ \sigma_{A,A*} \otimes \mathsf{id}_A) \circ (\mathsf{id}_A \otimes \epsilon_A) = \mathsf{id}_A$; $\pi_i = q_i^\dagger$ for the biproducts; and also such that the symmetric monoidal natural isomorphisms are unitary with respect to †.

In order to define the semantics of our language we will need some Cartesian constructs. An SCCCB does indeed come with a product structure, but since

this also corresponds with the coproduct structure it does not suit our purposes. At this stage we make the assumption that our classical machine has a finite number n of registers, each of which can contain only finite m values. The classical value space then is given as a function $n \to m$ (here identifying m with the set $\{0, \ldots, m-1\}$). In an SCCCB, given an object A we define $k.A$ as $A^{\oplus k}$ using the biproduct structure (in particular using it as a coproduct). We then define our classical state space as $m^n.I$ where I is the monoidal unit. Distributivity isomorphisms are present in any SCCCB, giving an isomorphism $a.A \otimes b.A \cong ab.A$.

Definition Given any SCCCB $\mathcal{C}$ we can construct the *classical subcategory* of $\mathcal{C}$ with objects of type $I \oplus \ldots \oplus I$ and arrows $n.I \to m.I$ are those defined from a function $f : \{1 \ldots n\} \to \{1 \ldots m\}$ in the obvious manner (permuting the coproduct possibilities, i.e. arrows $[q_{f(1)}, \ldots, q_{f(n)}]$).

Proposition 3.1 *This subcategory is distributive — that is, it has a terminal object, products and weak coproducts. Further, the canonical map* $[\mathsf{id} \times q_1, \mathsf{id} \times q_2] : (A \times B) + (A \times C) \to A \times (B + C)$ *has a left inverse* $\mathsf{dist} : A \times (B + C) \to (A \times B) + (A \times B)$ *so that* $\mathsf{dist}.[\mathsf{id} \times q_1, \mathsf{id} \times q_2] = \mathsf{id}$.

We need to address how to represent $\mathsf{DProb}(S \times H)$ as an object in our category. We can use $m^n.I$ to represent S using the classical subcategory. To represent a qubit, we define the qubit space $Q = I \oplus I$ where I is the monoidal unit (in **FdHilb**, this is the one-dimensional Hilbert space $\mathbb{C}$). H is given by the combined space of k qubits, and is given by the object $H = Q^{\otimes k}$. Thus the compound system representing our combined classical/quantum components is given by $X = H \otimes S$. Note that this is $m^n.I \otimes H \cong m^n.H$ (again using distributivity) and so this can also be viewed as the m^n-fold coproduct of H. This is explicitly representing the *quantum data + classical control* paradigm [Sel04b] since the classical control is determined entirely by which of the m^n components of the coproduct we are in, and then in each component we have a space H representing the quantum part of the structure. However, we in fact need to deal with probability distributions over states.

Since we have only a finite number (m^n) of classical states, an element of $\mathsf{DProb}(S \times H)$ can be represented as m^n sub-distributions on H. Thus, we need only then deal with a categorical representation of probability distributions on H. It is well known in the literature that a probability distribution on a Hilbert space H can be represented as a *mixed state* i.e. a ray in $H \otimes H^*$. Given a pure state u in H the corresponding mixed state is represented by $u \otimes \bar{u}$ — this looses no information, and we can then use weighted sums to represent probability distributions over these pure states [NC00]. We can also extend this functorially, mapping operators to the corresponding operators on the mixed state space.

It is possible to lift these ideas to our abstract setting, following ideas in [Sel07]. Given an SCCCB $\mathcal{C}$ we define $\mathbf{CPM}(\mathcal{C})$ to be a category with objects from $\mathcal{C}$, but morphisms $A \to B$ in $\mathbf{CPM}(\mathcal{C})$ are given by arrows $A \otimes A^* \to B \otimes B^*$ in $\mathcal{C}$. It is known that $\mathbf{CPM}(\mathcal{C})$ is also an strongly compact closed category, but it does not inherit the biproducts from $\mathcal{C}$. It is, however, monoid-enriched (by $+$, inherited from the biproducts in $\mathcal{C}$) — and we can define the biproduct completion $(-)^{\oplus}$ of a monoid-enriched category, where arrows are given by matrices [Sel07]. Thus we will give our semantics in the SCCCB $\mathbf{CPM}(\mathcal{C})^{\oplus}$.

We can define a functor $F : \mathcal{C} \to \mathbf{CPM}(\mathcal{C})$ taking an object H to $H \otimes H^*$ with $F(h) = h \otimes h_*$.

Given an SCCCB $\mathcal{C}$ we define our qubit space $Q = I \oplus I$ in $\mathcal{C}$ and derive H from Q as above. We then work in $\mathbf{CPM}(\mathcal{C})^{\oplus}$ where (probability distributions over) our quantum space is represented by $\langle H \rangle$ and the biproducts used for our classical state (as above) are the free biproducts. Thus $\mathsf{DProb}(S \times H)$ will be represented by $m^n.F(H)$.

3.2 Denotational Semantics

We now present the categorical axioms required to give corresponding denotational and operational semantics for our language above. We recall that the biproducts give us an addition monoid $(0, +)$ on hom-sets. A *scalar* is an endomorphism on the monoidal unit, and such a scalar s is *positive* if it can be factorised into $h \circ h^\dagger$. We can define multiplication of arrow s by scalar λ, denoted $\lambda \bullet s$.

Definition A *pre-quantum recursive category* is a strongly compact closed category $\mathcal{C}$ with biproducts such that $\mathbf{CPM}(\mathcal{C})$ is cpo-enriched. We require that composition and addition are continuous with respect to this ordering and that $0 = \bot$. We further require that the nonzero positive scalars are closed under addition, multiplication (composition), inverses and square roots. Finally we require that addition and multiplication are monotonic with respect to $\sqsubseteq$ on the positive scalars.

Note that the scalars in $\mathbf{CPM}(\mathcal{C})$ correspond precisely to the scalars in $\mathcal{C}$, and to scalars in $\mathbf{CPM}(\mathcal{C})^{\oplus}$. If s is a scalar in $\mathbf{C}$ then $F(s)$ is a positive scalar in $\mathbf{CPM}(\mathcal{C})$. This "squares" s. However we can also embed s in $\mathbf{CPM}(\mathcal{C})$ *directly* since $I \otimes I^* \cong I$. We write $G(s)$ for such a scalar, and can show that if s is positive then $G(s)$ is a completely positive arrow, so exists in $\mathbf{CPM}(\mathcal{C})$ [Chu07]. We extend the cpo structure on $\mathbf{CPM}(\mathcal{C})$ to $\mathbf{CPM}(\mathcal{C})^{\oplus}$ by componentwise comparison of arrows. We can show that addition, composition, multiplication etc are continuous in $\mathbf{CPM}(\mathcal{C})^{\oplus}$.

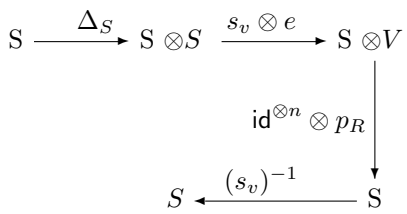
We have called the above a *pre*-quantum recursive category because although **Rel** is an instance, **FdHilb** is not. For this we require some weakening, but that requires some further work which we will tackle after giving full semantics for our language in a PQRC and showing that they correspond.

We will give meaning to any command as an arrow $X \to X$ in $\mathbf{CPM}(\mathcal{C})^{\oplus}$ where $X = m^n.F(H)$ using the free biproducts (here we identify $\mathcal{C}$ as a full subcategory of $\mathcal{C}^{\oplus}$). H is defined (in the PQRC $\mathcal{C}$) as $Q^{\otimes k}$ where k is the number of qubits and Q is the qubit space $I \oplus I$ (here using biproducts in $\mathcal{C}$ itself). Since $X = n^m.F(H)$ we know $X \cong S \otimes F(H)$ where $S = m^n.I$ and so (if we so desire) we can write an arrow $X \to X$ as $f \otimes g$ where $f : S \to S$ and $g : F(H) \to F(H)$. We now give denotational semantics as such arrows.

$\mathcal{D}[\![\mathtt{skip}]\!] = \mathsf{id}_X$
$\mathcal{D}[\![\mathtt{C}_1; \mathtt{C}_2]\!] = \mathcal{D}[\![C_2]\!] \circ \mathcal{D}[\![C_1]\!]$

Let $V = I^{\otimes m}$, i.e. the classical value space of an individual register. Given a variable $v \in n$ and a classical $e : S \to V$ (i.e. e is in the canonical classical

subcategory) we define the update map $[v \mapsto e] : S \to S$ as follows (p_R and s_v will be defined below).



Here we firstly clone the classical information with $\Delta_S : S \to S \otimes S$. We then apply $\mathsf{id} \otimes e$ taking us to $S \otimes V$, the old state space and the new variable. We then apply $s_V \otimes \mathsf{id}$, rearranging the old state space so that the variable we wish to alter is last (s_V is the unique symmetry isomorphism permuting the vth and final components of $S = V^{\otimes n}$). We then apply $\mathsf{id} \otimes \ldots \otimes \mathsf{id} \otimes p_R$ where $f : V \otimes V \to V$ is the right projection, forgetting the old value and keeping the new one.

We have used cloning ($\Delta_S : S \to S \otimes S$) and projection operators. This only works as a copying operation in the Cartesian classical category (which is where we are currently working) — no such (natural) copying operator can possibly exist in the full quantum category itself, see [Abr09].

We assume that any arithmetic expression e has a primitive (classical) denotation $[\![e]\!] : S \to V$, and define

$$\mathcal{D}[\![v{:=}e]\!] = [v \mapsto [\![e]\!]] \otimes \mathsf{id}_H$$

We next consider the conditional. Note that $\oplus$ is a coproduct, and we let $\mathsf{dist} : X \otimes (I \oplus I) \to X \oplus X$ denote the natural distributivity isomorphism. Once again we assume $[\![b]\!] : S \to I \oplus I$ as a primitive arrow. We can also define $\Delta : X \to X \otimes S$ that copies the classical part of the composite state (see [Chu07] for details).

$$\mathcal{D}[\![\texttt{if } b \texttt{ then } C_1 \texttt{ else } C_2]\!] = [C_1, C_2] \circ \mathsf{dist} \circ (\mathsf{id} \otimes [\![b]\!]) \circ \Delta$$

We now move on to iteration. This we define using the coproduct structure and cpo-enrichment in the expected way:

$$\mathcal{D}[\![\texttt{while } b \texttt{ do } C]\!] = \mathsf{lfp}[\lambda f : X \to X.([f \circ \mathcal{D}[\![C]\!], \mathsf{id}] \circ \mathsf{dist} \circ (\mathsf{id} \otimes [\![b]\!]).\Delta]$$

Finally we deal with the quantum cases. Given each binary unitary U we assume a primitive denotation $[\![U]\!] : Q \otimes Q \to Q \otimes Q$ We then define $\mathsf{app}_{u,x,y}$ to be the result of applying $[\![U]\!]$ at positions x and y, i.e. if $\sigma_{x,y}$ is the unique monoidal isomorphism $Q^{\otimes k} \to Q^{\otimes k}$ that sends the xth qubit to location 1 and the yth qubit to location 2, $\mathsf{app}_{u,x,y} = \sigma^{-1} \circ ([\![U]\!] \otimes \mathsf{id}_{Q^{k-2}}) \circ \sigma$.

$$\mathcal{D}[\![\texttt{apply } u \texttt{ in } q_1, q_2]\!] = \mathsf{id} \otimes F(\mathsf{app}_{u,q_1,q_2})$$

And for the measurement case, we define $P^i_j : H \to H$ in $\mathbf{C}$ as the arrow applying $q_j.\pi_j : Q \to Q$ to the ith qubit in parallel with identity operations on

the other qubit, in a similar manner to applying unitaries. A striking fact here is that we do not deal with the probability scalings, neither do we deal with normalisation of the resulting quantum states. This is because these two operations are inverses in our structure. This is fortunate indeed — representing normalisation from inside our category would involve calculating square roots, which would be a problem since everything is linear. An alternative perspective here is that we aren't normalising to 1, but rather to the probability at this point in the evaluation tree, cf. [Sel04b].

$$\mathcal{D}[\![\texttt{measure } q \texttt{ in } v]\!] = ([v \mapsto 0] \otimes F(P_0^q)) + ([v \mapsto 1] \otimes F(P_1^q))$$

This completes our definition of $\mathcal{D}[\![C]\!]$. It will be useful to define $\mathcal{D}'[\![C]\!] : \mathbf{CPM}(\mathcal{C})^{\oplus}(I, S) \times \mathcal{C}(I, H) \to \mathbf{CPM}(\mathcal{C})^{\oplus}(I, X)$ by $\mathcal{D}'[\![C]\!](s, \phi) = \mathcal{D}[\![C]\!] \circ (s \otimes F(\phi))$, taking a classical arrow and a quantum state and returning the resulting probability distribution after C is applied.

3.3 Operational Semantics

We now give operational semantics in this setting. A configuration is a combination (C, s, ϕ) where $s : I \to S$ is a classical arrow (i.e. q_i for some $i \leq k^m$) in $\mathbf{CPM}(\mathcal{C})^{\oplus}$, and $\phi : I \to H$ in the category $\mathbf{C}$. Thus s and ϕ represents "elements" of the respective objects.

Once again we define a one-step relation $\to$ between configurations, with each relation tagged with a probability, i.e. a positive scalar $I \to I$ in the category $\mathbf{C}$. Also it once again must be the case that the trees generated from this relation are finitely branching, and indeed they are — our reduction rules are as follows:

$$(v := e,\ s,\ \phi) \to^1 (\texttt{skip},\ [v \mapsto [\![e]\!]] \circ s,\ \phi)$$

$$(\texttt{skip};C,\ s,\ \phi) \to^1 (C,\ s,\ \phi)$$

$$\frac{(C_1,\ s,\ \phi) \to^p (C_1',\ s',\phi')}{(C_1;\ C_2,\ s,\ \phi) \to^p (C_1';\ C_2,\ s',\ \phi')}$$

$$\frac{[\![b]\!] \circ s = q_1}{(\texttt{if } b \texttt{ then } C_1 \texttt{ else } C_2,\ s,\ \phi) \to^1 (C_1,\ s,\ \phi)}$$

$$\frac{[\![b]\!] \circ s = q_2}{(\texttt{if } b \texttt{ then } C_1 \texttt{ else } C_2,\ s,\ \phi) \to^1 (C_2,\ s,\ \phi)}$$

$$\frac{[\![b]\!] \circ s = q_1}{(\texttt{while } b \texttt{ do } C,\ s,\ \phi) \to^1 (C\texttt{; while } b \texttt{ do } C,\ s,\ \phi)}$$

$$\frac{[\![b]\!] \circ s = q_2}{(\texttt{while } b \texttt{ do } C,\ s,\ \phi) \to^1 (\texttt{skip},\ s,\ \phi)}$$

$$(\texttt{apply } U \texttt{ to } q_1, q_2,\ s,\ \phi) \to^1 (\texttt{skip},\ s,\ \texttt{app}_{u,q_1,q_2} \circ \phi)$$

$$\texttt{measure}\ q\ \texttt{in}\ v,\ s,\ \phi) \rightarrow^{p_0^q(\phi)} (\texttt{skip},\ [v \mapsto 0] \circ s,\ \sqrt{p_0^q(\phi)^{-1}} \bullet P_q^0 \circ \phi)$$

$$\texttt{measure}\ q\ \texttt{in}\ v,\ s,\ \phi) \rightarrow^{p_1^q(\phi)} (\texttt{skip},\ [v \mapsto 1] \circ s,\ \sqrt{p_1^q(\phi)^{-1}} \bullet P_q^1 \circ \phi)$$

Measurements work as follows: once again we define projections $P_j : Q \to Q = q_j.\pi_j$ and extend this to $P_j^i : H \to H$ acting on qubit i in the natural way. For $\phi : I \to H$ we define the scalar $p_j^i(\phi)$ to be $\phi^\dagger.P_j^i.\phi$. Finally we note that the result of the measurement needs to be *normalised*, i.e. divided by the square root of the probability as in the concrete case. We will embed the probabilities into $\mathbf{CPM}(\mathcal{C})$ using G and will see that probabilities and normalisation cancel out, since $G(p_j^i(\phi)) \circ F(\sqrt{p_j^i(\phi)^{-1}} \bullet P_j^i \circ \phi) = F(P_j^i \circ \phi)$.

Once again by taking the reflexive transitive closure of our relation we obtain reduction trees labeled with probabilities, looking exactly as in the concrete semantics. We now need to use these operational semantics to form the semantic function, as above. This is where we use the **CPM** construction, for creating probabilistic weightings of states.

Given a command C and states $s, s' : I \to S$ and $\phi, \phi' : I \to H$ we define $\mathsf{Comp}(C, s, s', \phi, \phi')$ to be the set of all reductions $(C, s, \phi) \rightarrow^{p_1} \rightarrow \ldots \rightarrow^{p_n} (\texttt{skip}, s', \phi')$ and given such a $c \in \mathsf{Comp}(C, s, s', \phi, \phi')$ we define $p(c)$ to be Πp_i (note that multiplication of scalars $p : I \to I$ are commutative).

We then define

$$\mathsf{Prob}(C) : \mathbf{CPM}(\mathcal{C})^\oplus(I, S) \times \mathbf{CPM}(\mathcal{C})^\oplus(I, S) \times \mathcal{C}(I, H) \times \mathcal{C}(I, H) \to \mathbf{CPM}(\mathcal{C})^\oplus(I, I)$$

by

$$\mathsf{Prob}(C, s, s', \phi, \phi') = G(\Sigma\,\{p(c) | c \in \mathsf{Comp}(C, s, s', \phi, \phi')\})$$

and

$$\mathcal{O}[\![C]\!](s, \phi) = \Sigma_{s', \phi'}(\mathsf{Prob}(C, s, s', \phi, \phi') \bullet (s' \otimes F(\phi'))$$

using addition from the biproduct structure, summing over all (s', ϕ') such that $(C, s, \phi) \rightarrow\!\!\!\!\!\rightarrow (\texttt{skip}, s', \phi')$. Note that if (C, s, ϕ) never terminates to any solution this summation results in the zero distribution, i.e. the $\perp$ element; and we can use cpo-enrichment to calculate the infinite sums — details will be given below. This gives us $\mathcal{O}[\![C]\!] : \mathbf{CPM}(\mathcal{C})^\oplus(I, S) \times \mathbf{C}(I, H) \to \mathbf{CPM}(\mathcal{C})^\oplus(I, X)$.

Note that in passing from scalars in $\mathcal{C}$ to scalars in $\mathbf{CPM}(\mathcal{C})$ we embed via G rather than F. This is because we do not wish to square the scalars, and it allows normalisation and probability scalings to cancel out (as in [Sel04b]). We can show that the probabilities with which we annotate are indeed positive, so that $G(s)$ is a valid completely positive arrow in $\mathbf{CPM}(\mathcal{C})$ [Chu07].

We now give a more precise account of $\mathcal{O}(C)$. Let $\mathsf{Comp}_n(C, s, s', \phi, \phi')$ be the set of reduction paths from (C, s, ϕ) to $(\texttt{skip}, s', \phi')$ with at length at most n, and if c is such a path we write $p(c)$ for the probability weighting of that path, i.e. the product of all of the probabilities along this path (given by composition of scalars, which we know to be commutative and associative). We then define

$\mathcal{O}[\![C]\!]_n$ as a function from $\mathbf{CPM}(\mathcal{C})^{\oplus}(I,S) \times \mathcal{C}(I,H) \to \mathbf{CPM}(\mathcal{C})^{\oplus}(I,X)$. To define this we firstly define

$$\mathsf{Prob}_n(C,s,s',\phi,\phi') = G(\Sigma\,\{p(c)|c \in \mathsf{Comp}_n(C,s,s',\phi,\phi'))$$

This sum is defined using the biproduct structure, and this is a finite sum since the reduction tree we have will be a finitely branching one. We know that there will only be finitely many s', ϕ' such that $(C,s,\phi)\overset{*}{\to}(\mathtt{skip},s',\phi')$ in at most n steps. Hence we define

$$\mathcal{O}[\![C]\!]_n(s,\phi) = \Sigma\,\{\mathsf{Prob}_n(C,s,s',\phi,\phi') \bullet (s' \otimes F(\phi'))|(C,s,\phi)\overset{*}{\to}(\mathtt{skip},s',\phi')\}$$

summing over such (s',ϕ'). We note that functions $\mathbf{CPM}(\mathcal{C})^{\oplus}(I,S)\times\mathcal{C}(I,H) \to \mathbf{CPM}(\mathcal{C})^{\oplus}$ form a complete partial order, with pointwise ordering (since our underlying codomain hom-sets are cpo-enriched). Hence we define $\mathcal{O}[\![C]\!]$ to be the least upper bound of the chain $(\mathcal{O}[\![C]\!]_n)_n$ (so $\mathcal{O}[\![C]\!](s,\phi) = \bigsqcup(\mathcal{O}[\![C]\!]_n(s,\phi))$). We need to check that this is indeed a chain.

To do this, we need to show that $\mathcal{O}[\![C]\!]_{n+1}(s,\phi) \sqsupseteq \mathcal{O}[\![C]\!]_n(s,\phi)$. Since we know that

$$\mathsf{Comp}_n(C,s,s',\phi,\phi') \subseteq \mathsf{Comp}_{n+1}(C,s,s',\phi,\phi')$$

this amounts to requiring that if p is a probability (arising from a product of probabilities from the reduction tree) then $a + p \sqsupseteq p$ in the cpo-structure ordering. Our monotonicity assumption of + guarantees this.

3.4 Correspondence

Proposition 3.2 *For any command C, $\mathcal{D}'[\![C]\!](s,\phi) = \mathcal{O}[\![C]\!](s,\phi)$ for all ϕ and for classical s.*

Proof Full details given in [Chu07] pp53-57 together with supporting lemmas.

In the case that $C = \mathtt{skip}$ we have $\mathcal{D}'[\![C]\!](s,\phi) = \mathcal{D}[\![C]\!] \circ (s \otimes F(\phi)) = \mathsf{id}.(s \otimes F(\phi)) = s \otimes F(\phi)$. Also $(C,s,\phi)\overset{*}{\to}^1(\mathtt{skip},s,\phi)$ and so $\mathsf{Comp}(C,s,s',\phi,\phi')$ is nonempty only if $s = s'$ and $\phi = \phi'$ and the set $\mathsf{Comp}(C,s,s,\phi,\phi)$ contains a single empty computation branch whose probability is 1, the empty product. As such $\mathsf{Prob}(s,s,\phi,\phi) = 1 = \mathsf{id}$ and so $\mathcal{O}[\![C]\!](s) = 1.(s \otimes F(\phi)) = s \otimes F(\phi) = \mathcal{D}'[\![C]\!](s,\phi)$ as required.

We secondly consider the case $C = C_1; C_2$. We need to show that $\mathcal{O}[\![C]\!](s,\phi) = \mathcal{D}'[\![C]\!](s,\phi)$. By definition the right hand side is $\mathcal{D}[\![C]\!] \circ (s \otimes F(\phi)) = \mathcal{D}[\![C_2]\!] \circ \mathcal{D}[\![C_1]\!] \circ (s \otimes F(\phi)) = \mathcal{D}[\![C_2]\!] \circ \mathcal{D}'[\![C_1]\!](s,\phi) = \mathcal{D}[\![C_2]\!] \circ \mathcal{O}[\![C_1]\!](s,\phi)$ by inductive hypothesis. Hence it suffices to show that $\mathcal{O}[\![C]\!](s,\phi) = \mathcal{D}[\![C_2]\!].\mathcal{O}[\![C_1]\!](s,\phi)$.

Let (C_1,s,ϕ) reduce to $(\mathtt{skip},s_1,\phi_1)\ldots(\mathtt{skip},s_n,\phi_n)$ with probabilities $p_1 \ldots p_n$ (note that this covers all cases since if (C_1,s,ϕ) does not terminate then $n = 0$. In this case our final probability is the empty sum 0, which coincides with $\perp$). Then $\mathcal{O}[\![C_1]\!](s,\phi) = \Sigma(p_i \bullet (s_i \otimes F(\phi_i)))$. Hence our expression above is $\mathcal{D}[\![C_2]\!] \circ \Sigma(p_i \bullet (s_i \otimes F(\phi_i)))$ which is $\Sigma(p_i \bullet \mathcal{D}[\![C_2]\!] \circ (s_i \otimes F(\phi_i)))$ by distributivity and properties of scalars. This is $\Sigma(p_i \bullet \mathcal{D}'[\![C_2]\!](s_i,\phi_i))$ which is once again $\Sigma(p_i \bullet \mathcal{O}[\![C_2]\!](s_i,\phi_i))$ by inductive hypothesis. Hence it suffices to show that $\mathcal{O}[\![C]\!](s,\phi)$ is this expression.

Well let (C_2,s_i,ϕ_i) reduce to $(\mathtt{skip},s_{1i},\phi_{1i})\ldots(\mathtt{skip},s_{m_i i},\phi_{m_i i})$ with probabilities $p_{1i},\ldots,p_{m_i i}$. We note that then by studying the operational semantics that (C,s) reduces to

$$(s_{11}, \phi_{11}) \ldots, (s_{m_1 1}, \phi_{m_1 1}), \ldots,$$
$$(\mathrm{s}_{1n}, \phi_{1n}), \ldots, (s_{m_n n}, \phi_{m_n n})$$

with probabilities

$$p_{11}.p_1, \ldots, p_{m_1 1}.p_1, \ldots, p_{1n}.p_n, \ldots, p_{m_n n}.p_n$$

It follows then that $\mathcal{O}[\![C]\!](s, \phi) = \Sigma\Sigma p_{ij} \bullet p_j \bullet (s_{ij} \otimes F(\phi_{ij})$. By definition, this is $\Sigma(p_i \bullet \mathcal{O}[\![C_2]\!](s_i, \phi_i))$ as required. The above can be extended to the infinite case just using an infinite sum (using continuity together with finite distributivity to get infinite distributivity, etc).

In the case that $C = \mathrm{v} := \mathrm{e}$ then we need to show that $\mathcal{O}[\![C]\!](s, \phi) = \mathcal{D}'[\![C]\!](s, \phi) = ([v \mapsto [\![e]\!]] \otimes \mathsf{id}) \circ (s \otimes F(\phi)) = [v \mapsto [\![e]\!]] \circ s \otimes F(\phi)$. This clearly holds since (C, s, ϕ) reduces only to $(\mathtt{skip}[v \mapsto [\![e]\!]] \circ s, \phi)$ with probability 1.

The case $C = \mathtt{apply}\ u\ \mathtt{to}\ q_1, q_2$ is similar. We need to show that $\mathcal{O}[\![C]\!](s, \phi) = \mathcal{D}'[\![C]\!](s, \phi) = (id \otimes F(app_{u,x,y})) \circ (s \otimes F(\phi)) = s \otimes F(\mathsf{app}_{u,x,y}) \circ F(\phi) = s \otimes F(\mathsf{app}_{u,x,y} \circ \phi)$. This does indeed hold again by definition of operational semantics since (C, s, ϕ) reduces uniquely to $(\mathtt{skip}, s, \mathsf{app}_{u,x,y} \circ \phi)$

We now consider measurement $C =$ measure q in v. Firstly $\mathcal{D}'[\![C]\!](s, \phi) = \Sigma_j([v \mapsto j] \otimes F(P_j^q)) \circ (s \otimes F(\phi)) = \Sigma_j([v \mapsto j] \circ s \otimes F(P_j^q \circ \phi))$. Secondly $\mathcal{O}[\![C]\!](s, \phi) = \Sigma_j(G(p_j^q(\phi)) \bullet ([v \mapsto j] \circ s \otimes F(\sqrt{p_j^q(\phi)}^{-1} \bullet P_j^q \circ \phi))) = \Sigma_j(G(p_j^q(\phi)) \bullet ([v \mapsto j] \circ s \otimes F(\sqrt{p_j^q(\phi)}^{-1}) \bullet F(P_j^q.\phi))) = \Sigma_j(G(p_j^q(\phi)) \bullet F(\sqrt{p_j^q(\phi)}^{-1}) \bullet ([v \mapsto j]\ .s \otimes F(P_j^q.\phi)))$. It remains only to show that $G(p_j^q(\phi)) \bullet F(\sqrt{p_j^q(\phi)})^{-1} = 1$ (i.e. that the probabilistic weight and the normalisation do indeed cancel out) and in [Chu07] we show that $G(s) = F(\sqrt{s})$ and so $G(p_j^q(\phi)) \bullet F(\sqrt{p_j^q(\phi)})^{-1} = F(\sqrt{p_j^q(\phi)} \bullet F(\sqrt{p_j^q(\phi)})^{-1} = \mathsf{id}$, as required.

For conditional, we reduce the problem to showing that $\mathcal{D}[\![C]\!] \circ (s \otimes F(\phi)) = \mathcal{D}[\![C_i]\!] \circ (s \otimes F(\phi))$. We show this by exploiting classical copying and coproduct equations, naturality, distributivity etc. See [Chu07] for details.

For the `while` case we show separately that $\mathcal{O}[\![C]\!] \sqsupseteq \mathcal{D}'[\![C]\!]$ and $\mathcal{O}[\![C]\!] \sqsubseteq \mathcal{D}'[\![C]\!]$. To show the former we show that $\mathcal{D}'[\![C]\!]$ is the least fixpoint of the function $g = \lambda f.\lambda(s, \phi).\ \mathsf{if}\ [\![b]\!] \circ s = q_1\ \mathsf{then}\ f(\mathcal{D}[\![C_1]\!] \circ (s \otimes F(\phi)))\ \mathsf{else}\ (s \otimes F(\phi))$ and that $\mathcal{O}[\![C]\!]$ is also a fixpoint. To show the latter, we demonstrate that for all n there is some m such that $\mathcal{O}[\![C]\!]_n(s, \phi) \sqsubseteq (h^m \bot)(s \otimes F(\phi))$ where h is the endofunction above (given in the definition of the while case denotational semantics). From this it follows that for all n, $\mathcal{O}[\![C]\!]_n \leq \mathcal{D}[\![C]\!]$ and so the limit of the chain $\mathcal{O}[\![C]\!]_n$ is at most $\mathcal{D}[\![C]\!]$, i.e. precisely that $\mathcal{O}[\![C]\!] \sqsubseteq \mathcal{D}[\![C]\!]$. ∎

4 Categories of Superoperators

The above formulation requires the category $\mathbf{CPM}(\mathcal{C})$ to be cpo-enriched. While the category **Rel** does satisfy this requirement, **FdHilb** does not. **FdHilb** is

only cpo-enriched when considering only trace-decreasing operators. For example, the ordering given for **FdHilb** in [Sel04b] when restricted to positive scalars gives the standard ordering positive scalars, which clearly does not admit least upper bounds for arbitrary ω-chains.

Definition Any "element" in the CPM category $I \to A \otimes A^*$ can be written as the lambda abstraction of a map $A \to A$. Given such an $s = \Lambda(f)$ then we define $\mathsf{tr}(s)$ to be the scalar $\epsilon_A \circ (id_{A^*} \otimes f) \circ \eta_A : I \to I$ (by compact closure, this is equal to $\epsilon_A \circ s$).

This coincides with treatment given in e.g. [Coe05]. Further details can be found in e.g. [AC04] and we note that this is a direct abstraction of the linear algebraic trace operator. We now generalise this to tuples of **CPM**-objects using the free biproduct closure.

Definition Let $f : I \to A$ where A is some object in $\mathbf{CPM}(\mathcal{C})^{\oplus}$. Then since objects in $\mathbf{CPM}(\mathcal{C})^{\oplus}$ are tuples $\langle A_1, \ldots A_n \rangle = A_1 \oplus ... \oplus A_n$, we have $f = \langle f_1, \ldots, f_n \rangle$ with $f_i : I \to A_i$. Since A_i is an object in $\mathbf{CPM}(\mathcal{C})$ and so is of the form $X \otimes X^*$ we can take the trace of each f_i. We then define $\mathsf{tr}(f) = \Sigma(\mathsf{tr}(f_i))$.

Definition A map $f : A \to B$ in $\mathbf{CPM}(\mathcal{C})^{\oplus}$ is said to be *trace-decreasing* if for any $s : I \to A$ we have $\mathsf{tr}(f \circ s) \sqsubseteq \mathsf{tr}(s)$.

We note that trace-decreasing arrows are closed under composition etc. and clearly the identity is trace decreasing.

Definition The category $\mathbf{SUP}(\mathcal{C})^{\oplus}$ is the lluf subcategory of $\mathbf{CPM}(\mathcal{C})^{\oplus}$ restricted to trace-decreasing arrows (*superoperators* in the literature e.g. in [Sel04b]).

The notation may be slightly misleading here, since $\mathbf{SUP}(\mathcal{C})^{\oplus}$ is not meant to refer to the biproduct completion of $\mathbf{SUP}(\mathcal{C})$ (the category of completely positive trace-decreasing maps — in fact this does not make sense, as $\mathbf{SUP}(\mathcal{C})^{\oplus}$ is not necessarily closed under addition).

We now have identified three principal subcategories of $\mathbf{CPM}(\mathcal{C})^{\oplus}$ of interest — the canonical classical subcategory, the full subcategory $\mathbf{CPM}(\mathcal{C})$ of single-tuple objects, and $\mathbf{SUP}(\mathcal{C})^{\oplus}$. Our requirement, then, is that $\mathbf{CPM}(\mathcal{C})$ is order-enriched as before, and that the derived ordering in $\mathbf{CPM}(\mathcal{C})^{\oplus}$ is complete for homsets of $\mathbf{SUP}(\mathcal{C})^{\oplus}$.

Definition A *quantum recursive category* consists of a strongly compact closed category $\mathcal{C}$ with biproducts with an order relation on $\mathbf{CPM}(\mathcal{C})$ that is complete when extended to homsets of $\mathbf{SUP}(\mathcal{C})^{\oplus}$. We require that composition and copairing are continuous with respect to this ordering and $0 = \bot$. We also require that the nonzero positive scalars are closed under addition, multiplication, inverses and square roots. Finally we require that addition and multiplication are monotonic with respect to $\sqsubseteq$ on the positive scalars.

We can show that trace-decreasing completely positive maps are closed under composition and we can show that trace-decreasing completely positive maps are closed under copairing [Chu07].

Finally we note that any pre-quantum recursive category is also a quantum recursive category, provided the limit of a chain of trace-decreasing arrows is trace-decreasing.

We can now show that our semantics are still well-defined. Firstly we tackle the denotational semantics — since our order completion now only applies to superoperators we require that

Proposition 4.1 *1. For any non-*`while` *command* C, $\mathcal{D}[\![C]\!] : X \to X$ *is trace decreasing. 2. The function* $\lambda f : X \to X.([f \circ \mathcal{D}[\![C]\!], \mathsf{id}] \circ \mathsf{dist} \circ (\mathsf{id} \otimes [\![b]\!]).\Delta$ *is continuous and maps trace-decreasing arrows to trace-decreasing arrows.*

This is shown in [Chu07]. For our operational semantics we also now require that ϕ is normalised in our configurations, i.e. $\mathsf{tr}(F(\phi)) = 1$. We can then show the following (see [Chu07] for details)

Proposition 4.2 *If* $(C, s, \phi) \to (C', s', \phi')$ *and* ϕ *is normalised, then so is* ϕ'.

Proposition 4.3 $p_0^q(\phi) + p_1^q(\phi) = 1$

Proposition 4.4 $tr(\mathcal{O}[\![C]\!]_n(s, \phi)) \sqsubseteq 1$.

Hence $\mathcal{O}[\![C]\!]_n(s, \phi)$ is indeed trace-decreasing and so the infinite sum makes sense as a limit of a cpo-process in the $\mathbf{SUP}(\mathcal{C})^{\oplus}$ subcategory.

The proof of the correspondence theorem is largely unchanged for this version of the semantics (of course now it is only valid for normalised ϕ) — the only alteration is that the function g in the while case (for which $\mathcal{D}'[\![C]\!]$ is the least fixed point) is now an endofunction on the space $\mathbf{CPM}(\mathcal{C})^{\oplus}(I, S) \times \mathcal{C}(I, H)^* \to \mathbf{SUP}(\mathcal{C})^{\oplus}(I, X)$; where $\mathcal{C}(I, H)^*$ consists of normalised ϕ in $\mathcal{C}(I, H)$.

4.1 Examples of Quantum Recursive Categories

To recall, we define our semantics in an SCCCB $\mathcal{C}$ such that $\mathbf{CPM}(\mathcal{C})$ is order-enriched and this order is complete for $\mathbf{SUP}(\mathcal{C})^{\oplus}$, satisfying some further laws regarding interaction and scalars. We now need to check that these requirements are reasonable, which to us means that they are satisfied by **FdHilb** and **Rel**.

Firstly to see that **FdHilb** provides a valid model, we appeal to results from [Sel04b]. It is well-known that **FdHilb** is a strongly compact closed category with biproducts. For the order relation on $\mathbf{CPM}(\mathbf{FdHilb})$ we use the *Lowner partial order* on completely positive maps [Sel04b]. This states that $A \sqsubseteq B$ if and only if $B - A$ is a positive matrix. This is a partial order that has a bottom element (which is 0 as we require) and in fact supports least upper bounds for trace-decreasing completely positive maps. Furthermore, the inherited ordering is complete for homsets of $\mathbf{SUP}(\mathbf{FdHilb})^{\oplus}$ as we require — a proof of this is given in [Sel04b], although using different notation (the $\mathbf{Q}$ of [Sel04b] is our $\mathbf{SUP(FdHilb)}^{\oplus}$). Composition and copairing are continuous.

We note that a scalar a is positive if it it can be expressed $b.b^{\dagger}$, i.e. if it can be written in the form $\Sigma a_i.a_i^{\dagger}$ for complex numbers a_i. Scalars in **FdHilb** are complex numbers, and the complex numbers expressible in this form are precisely the positive reals. Thus the positive scalars are indeed the positive real numbers. As such, positive (nonzero) scalars are closed under addition, multiplication, inverses and square roots. Furthermore then for scalars b and a, $b - a$ is positive if and only if $b \geq a$ in the usual ordering on real numbers —

and so composition and addition are indeed monotonic. Note that the trace-decreasing positive scalars are those in $[0, 1]$ i.e. our valid concrete probabilities.

We can show that the categorical setting $\mathcal{C} = \mathbf{FdHilb}$ coincide with the concrete semantics given above (see [Chu07]).

We now show that **Rel** too satisfies our requirements. As mentioned, **Rel** is a strongly compact closed category with biproducts. We need to endow **CPM(Rel)** with an order-relation and show that is complete for trace-decreasing maps. There is a natural ordering on homsets of **Rel** itself — namely set inclusion. We can show that this partial order is a cpo on **CPM(Rel)**. Further we can show that

Proposition 4.5 $\perp$ *is completely positive and the union of a chain of completely positive relations is also completely positive.*

So it seems that **CPM(Rel)** is complete as it is, and so we don't in particular need to consider trace decreasing operators. To check further conditions, we note that $\perp$ is indeed the zero map (i.e. the empty set) and composition and addition is continuous (the sum of two arrows in **Rel** is their union).

A scalar in **Rel** is an arrow $I \to I$, i.e. a relation between I and I, where $I = \{*\}$. There are two such relations — the empty relation and the relation relating $*$ with $*$ — we shall label these 0 and 1 respectively. A relation is positive if it is symmetric and partial reflexive ($xRy \Rightarrow xRx$) [Sel07]. Both scalars 0 and 1 in **Rel** satisfy this property (0 satisfies both vacuously, and 1 is clearly both symmetric and reflexive). Thus the positive scalars are indeed closed under addition. Likewise the scalars have square-roots ($1 = 1.1$, $0 = 0.0$) and also non-zero elements admit inverses ($1 = 1.1$). Finally, in the ordering we have $1 \sqsupset 0$ and so addition and multiplication are indeed monotonic, since $1 + 1 = 1$ and $1.1 = 1$. So **Rel** is a pre-quantum recursive category.

5 Further Directions

The results described here are given in (much) more detail in Churchill's master's thesis, *Abstract Semantics for a Simple Quantum Programming Language* [Chu07]. In this thesis full proofs are given for all of the results. It also extends the work here by considering logical semantics — we define predicates on the state space of our system and define a modal operator for programs, given meaning by our denotational semantics. This logic can only reason about the classical part of the system; but can hence indirectly reason about the quantum part via the use of measurements. We also explore an abstract version of the logical semantics using subobjects and pullbacks. In addition, we provide a worked example (in the concrete case) of the Deutch-Josza algorithm and give much more background information.

As mentioned in the original talk [Abr04] we could make the language slightly more structured, introducing types etc, making the language more functional. It is these typed languages that fit into ideas of category theory more easily, and would perhaps give a different flavour to our language and its semantics. Fundamentally, however, these semantics will still be the same: using the cpo-enrichment for recursion, biproducts and CPM construction to represent the spaces in general etc; it's just effectively the values of k, m and n could be

varied. Of course, once this has been done, we could extend our ideas to other programming language ideas: the main issue that our framework above does not currently allow is that of infinite (classical) datatypes since we represent our space by a finite biproduct.

The categories that we use to model our programs require stronger structure than the basic SCCCBs (such as cpo-enrichment and its interaction with the other features, as well as the logical semantic requirements). In [Chu07], we have investigated to some degree which of the logical semantic requirements follow automatically from the SCCCB structure; independence of all of our requirements on the category could be a further area of study to investigate.

Our language deals with a single, independent quantum system. A further area of research is the delicate interaction between quantum systems and ideas of concurrency; and seeing if we can model these ideas into our language. Also, from a practical point of view, we have assumed in the above very idealised hardware and have assumed away realistically necessary ideas such as error-correction. Seeing how these ideas fit into our framework could also be another potential direction.

References

[Abr04] S. Abramsky. A cook's tour of a simple quantum programming language. Lecture slides, 2004.

[Abr09] S. Abramsky. No-Cloning In Categorical Quantum Mechanics. *ArXiv e-prints*, October 2009.

[AC04] Samson Abramsky and Bob Coecke. A categorical semantics of quantum protocols. *Logic in Computer Science, Symposium on*, 0:415–425, 2004.

[AC05] Samson Abramsky and Bob Coecke. Abstract physical traces. *Theory and Applications of Categories*, 14:111, 2005.

[BBC+93] Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K. Wootters. Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Phys. Rev. Lett.*, 70(13):1895–1899, Mar 1993.

[Chu07] M. Churchill. Abstract semantics for a simple quantum programming language. Master's Thesis, University of Oxford, October 2007.

[Coe05] B. Coecke. Kindergarten Quantum Mechanics. *ArXiv Quantum Physics e-prints*, October 2005.

[JS91] A. Joyal and R. Street. Advances in mathematics. In *The geometry of the tensor calculus I*, pages 55–112. Springer-Verlag, 1991.

[NC00] M. Nielsen and I. Chuang. Quantum computation and quantum information, 2000.

[Sel04a] P. Selinger. A brief survey of quantum programming languages. pages 61–69, 2004.

[Sel04b] P. Selinger. Towards a quantum programming language. *Mathematical Structures in Computer Science*, 14(04):527–586, 2004.

[Sel07] Peter Selinger. Dagger compact closed categories and completely positive maps: (extended abstract). *Electronic Notes in Theoretical Computer Science*, 170:139 – 163, 2007. Proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL 2005).